REMARKS

GENERALLY

A current and Non-final Office Action is dated 04/11/2007. In this current Office Action, claims 1-59 were examined, and claims 1-59 were rejected.

With this current Reply, no claims are added, but claim 50 is canceled without prejudice. Hence, claims 1-49 and 51-59 are now presented for examination.

SUMMARY OF CURRENT OFFICE ACTION

The current Office Action noted that the information cited on one Information Disclosure Statement (IDS) (filed on March 21, 2006) was not considered because a legible copy of each item was not submitted.

The current Office Action objected to claims 40 and 53.

The current Office Action rejected claims 2-4, 6-9, 20-22, 29-31, 36-39, and 48-59 under 35 U.S.C. §112, second paragraph.

The current Office Action rejected claims 1-33 and 48-59 under 35 U.S.C. §101.

The current Office Action rejected claims 1, 10, 11, 13-19, 23-30, 32, 33, 48-50, 52-56, 58, and 59 under 35 U.S.C. §102(b).

The current Office Action rejected claims 2-9, 12, 20-22, 31, 34-47, 51, and 57 under 35 U.S.C. §103(a).

RESPONSE TO NOTATION REGARDING INFORMATION DISCLOSURE STATEMENT

It is believed that a legible copy of each item listed for the 21 March 2006 IDS was included in the original submission. Nevertheless, the PTO-1449 that was submitted on 21 March 2006 and another legible copy of each item listed thereon accompany this Reply.

RESPONSE TO CLAIM OBJECTIONS

The objected to claims 40 and 53 have been amended in manners believed to overcome the instituted objections.

Accordingly, withdrawal of the claim objections is hereby respectfully requested.

RESPONSE TO REJECTIONS UNDER 35 U.S.C. § 112

The current Office Action rejected claims 2-4, 6-9, 20-22, 29-31, 36-39, and 48-59 under 35 U.S.C. §112, second paragraph.

The asserted indefiniteness "for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention" is hereby respectfully traversed generally. Nevertheless, to facilitate prosecution of the instant Patent Application and to expedite its ultimate allowance as a U.S. Patent, certain claims have been amended in manners believed to overcome the instituted 35 U.S.C. §112, second paragraph, rejections.

Moreover, the rejections with respect to claims 29, 30, 31, 48, and 53 are specifically traversed. With regard to claims 29, 30, and 31, it is respectfully submitted that the "relative" terminology is internally linked so as to make it definite. One of ordinary skill in the art, in conjunction with the disclosure of the instant Paten Application, would know what is meant by "level of abstraction" and would understand which of two layers is relatively lower and which is relatively higher. With regard to claim 48, it is respectfully submitted that the allegedly "relative" terminology is internally linked so as to make "simpler", "increasingly complex", "progression", etc. definite within the claim. Similarly, with regard to claim 53, "simplicity" as a concept that is considered in an evaluation would be understood by one of ordinary skill in the art, at least in conjunction with the disclosure of the instant Paten Application.

Accordingly, withdrawal of the 35 U.S.C. §112 rejections is hereby respectfully requested.

RESPONSE TO REJECTIONS UNDER 35 U.S.C. § 101

The current Office Action rejected claims 1-33 and 48-59 under 35 U.S.C. §101. The propriety and legitimacy of the 35 U.S.C. §101 rejections is hereby respectfully traversed. Nevertheless, to facilitate prosecution of the instant Patent Application and to expedite its ultimate allowance as a U.S. Patent, certain claims have been amended in manners believed to overcome the instituted 35 U.S.C. §101 rejections.

Claims 1, 15, and 48 have been amended to recite: **realizing the derived API in one or more processor-accessible storage media.**

Claim 49 has been amended to address if the condition is met or is not met and to recite: that certain components/types are **realized in one or more processor-accessible storage media.**

Accordingly, withdrawal of the 35 U.S.C. §101 rejections is hereby respectfully requested.

ARGUMENT(S) AGAINST THE 35 U.S.C. §§ 102(b) AND 103(a) REJECTIONS AND ARGUMENTS FOR CLAIM PATENTABILITY

Generally, the current Office Action rejected claims 1, 10, 11, 13-19, 23-30, 32, 33, 48-50, 52-56, 58, and 59 under 35 U.S.C. §102(b) and rejected claims 2-9, 12, 20-22, 31, 34-47, 51, and 57 under 35 U.S.C. §103(a).

Specifically, the current Office Action reads as follows at the noted paragraph numbers (with any highlighting removed in this Reply):

> 11. Claims 1, 10, 11, 13-19, 23-30, 32, 33, 48-50, 52-56, 58, and 59 are rejected under 35 U.S.C. 102(b) as being anticipated by Burger et al. (US 5,097,533).
>
> 13. Claims 2-4 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burger et al. (US 5,097,533) in view of Hayes (US 6,006,279).
>
> 14. Claims 5-9, 12, 20-22, 31, 34, 35, 37-45, and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burger et al. (US 5,097,533) in view of Corrie, Jr. et al. (US 5,495,571).
>
> 15. Claim 36 is rejected under 35 U.S.C. 103(a) as being unpatentable over Burger et al. (US 5,097,533) in view of Corrie, Jr. et al. (US 5,495,571) as applied to Claim 34 above, and further in view of Hayes (US 6,006,279).
>
> 16. Claims 46, 51, and 57 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burger et al. (US 5,097,533).

The current Office Action therefore rejects the claims using one or more of Burger et al., Hayes, and Corrie, Jr. et al.

Burger et al. is apparently directed to a "System and method for interfacing computer application programs written in different languages to a software system" (the Title). The Abstract of Burger et al. reads as follows:

> A support system and method for interfacing of computer application programs written in a plurality of languages to a software system such as a database manager of the like. A plurality of generic application program interfaces or entry points are defined having a corresponding plurality of parameters in a consistent form required by the system to execute functions. The parameters are transformations of like parameters associated with the application programs which call the APIs. Processor states corresponding to threads in the application programs are stored in a table shared by the generic APIs. Upon return from the call and execution of the system function, processor state is restored and control returned to the application program. Necessity for separate entry points for applications written in each different supported language is thereby avoided as well as associated increased development effort, maintenance, and support.

Hayes is apparently directed to a "Plug-in module host framework" (the Title). The Abstract of Hayes reads as follows:

> Computer-executable process steps stored on a computer-readable medium to provide an application programming interface (API) to a client application for communicating between the client application and a plug-in module. The API includes a menu-create routine which searches for plug-in modules, which enters information regarding a found plug-in module into a structure, the structure being used to create and to display a plug-in menu of found plug-in modules, an "about" menu-create routine which enters "about" plug-in information into an "about" structure, the "about" structure being used to create and to display a plug-in "about" menu, a menu-enable routine, a menu-disable routine and an invoking routine which permits the client application to utilize a selected plug-in module.

Corrie, Jr., et al. is apparently directed to a "Method and system for performing parametric testing of a functional programming interface" (the Title). The Abstract of Corrie, Jr., et al. reads as follows:

The present invention provides a method and system for performing parametric testing of a functional programming interface. Parametric testing of a function verifies that the function performs as expected when a valid or an invalid parameter is passed to the function. To perform parameter testing on a function, the present invention receives as input prototype information for the function, and then formulates a testing plan. The invention tests the function according to the testing plan. The testing plan specifies a list of invalid and valid values for each parameter of the function. The invention repeatedly invokes the function, each time passing the function various combinations of invalid and valid values. The function passes the test when (1) the function does not return an error code for any combination of valid parameter values and (2) the function returns an error for any combination of parameter values that include an invalid parameter value.

None of Burger et al., Hayes, or Corrie, Jr. et al. are directed to *deriving* an application programming interface (API). Furthermore, it is respectfully submitted that none of Burger et al., Hayes, or Corrie, Jr. et al. describe, teach, or otherwise render unpatentable **deriving the API from the core scenario responsive to the plurality of code samples** (from example claim 1) in conjunction with the other claimed elements.

The rejection of independent claim 1 from the current Office Action reads on Page 10 as follows (with any highlighting removed):

> As per Claim 1, Burger et al. disclose:
>
> - preparing a plurality of code samples for a core scenario, each respective code sample of the plurality of code samples corresponding to a respective programming language of a plurality of programming languages (see Column 5: 30-36, "... the invention generalizes existing entries to a plurality of applications 16 written in any of a number of predetermined languages. "); and
>
> - deriving the API from the core scenario responsive to the plurality of code samples (see Column 5: 36-43, " ... the function of the generic APIs 14 of

the present invention is to transform parameters received in a number of different formats determined by the particular language in which the application 16 is written into forms expected by the DBM APIs 12 and operating system APIs 20 as dictated by the particular procedural language in which these APIs 12 and 20 are written. ").

In the current Office Action as reproduced above, the rejection of claim 1 cites to Column 5, Lines 30-43 of Burger et al. Burger et al. reads at Column 5, Lines 13-43 as follows:

> Still referring to FIG. 1, it will be recalled that one problem associated with prior systems was that the operating system 18 and program services 10 assumed that application 16 would be written in the same language and accordingly application program interfaces such as the APIs 20 to the operating system 18 and the APIs to the particular program services 10 such as database manager APIs 12 were interfaces written to a specific language. Thus, the APIs 12 and 20 were not generic and did not expect to be called from applications 16 written in more than one programming language. Thus, for example, if the database manager program services 10 was written in the C-Language, the associated DBM APIs 12 expected calls from the applications 16 such as a call to create or delete a database to conform to calls in the C-Language and when such calls from the applications 16 were in a different language, the applications 16 would not execute properly. In the present invention, however, a plurality of generic APIs 14 are provided to these pre-existing entry points of the DBM APIs 12 and operating system APIs 20 whereby the invention generalizes existing entries to a plurality of applications 16 written in any of a number of predetermined languages. *Thus the function of the generic APIs 14 of the present invention is to transform parameters received in a number of different formats determined by the particular language in which the application 16 is written into forms expected by the DBM APIs 12 and operating system APIs 20 as dictated by the particular procedural language in which these APIs 12 and 20 are written.*
>
> (*italicized* emphasis added by this Reply)

As is apparent from the above excerpt of Burger et al., Burger et al. is directed to a transformation of parameters by "generic APIs 14" so as to

accommodate the parameter formats "expected by the DBM APIs 12 and operating system APIs 20".

Hence, it is respectfully submitted that Burger et al. does not teach *deriving* an API.

Moreover, it is respectfully submitted that Corrie, Jr., et al. is directed to standard parameter testing on functions. It is not related to **performing one or more usability studies on the API utilizing a plurality of developers** (e.g., from dependent claim 5).

Thus, it is respectfully submitted that no art of record, either alone or in any combination, anticipates or renders obvious the elements of independent claims 1, 15, 34, 47, 48, and 49. Moreover, although each of the pending dependent claims also includes additional element(s) militating toward allowability, they are allowable at least for the reasons given above in connection with the independent claims.

Accordingly, it is respectfully requested that the §102 rejections and §103(a) rejections be withdrawn.

CONCLUSION

It is respectfully submitted that all of pending claims 1-49 and 51-59 are allowable. Consequently, allowance of claims 1-49 and 51-59 is hereby respectfully requested.

Respectfully Submitted,

Date: 10 Sept 2007

By: Keith W. Saunders

Keith W. Saunders
Reg. No. 41,462
(509) 324-9256 x238